En esta capítulo exploraremos la construcción libre de una categoría monoidal desde una mirada combinatoria con el uso de diagramas, junto a la clase de lenguajes que se pueden generar. Posteriormente, utilizaremos la definición de lenguajes recursivamente enumerables y su modelo de cómputo, las máquinas de Turing, para mostrar que estos son exactamente los mismos que los anteriores.

0.1. Categorías monoidales

Definición 1. Una signatura monoidal G es una signatura de la siguiente forma:

$$G_0^* \xleftarrow{\texttt{dom}} G_1 \xrightarrow{\texttt{cod}} G_0^*$$

G es finita si G_1 es finita. Un morfismo de signaturas monoidales $\varphi : G \to \Gamma$ es un par de funciones $\varphi_0 : G_0 \to \Gamma_0$ y $\varphi_1 : G_1 \to \Gamma_1$ tal que el siguiente diagrama conmuta:

$$\begin{tikzcd}
G_0^* \arrow[d, "\varphi_0"] & G_1 \arrow[l, "\texttt{cod}"'] \arrow[r, "\texttt{dom}"] \arrow[d, "\varphi_1"] & G_0^* \arrow[d, "\varphi_0"] \\
\Gamma_0^* & \Gamma_1 \arrow[l, "\texttt{cod}"'] \arrow[r, "\texttt{dom}"] & \Gamma_0^*
\end{tikzcd}$$

Con tales morfismos, las signaturas monoidales forman la categoría **MonSig**.

Definición 2. Una categoría monoidal estricta $(\mathcal{C}, \otimes, 1)$ es una categoría $\mathcal{C}$ con un bifuntor[1] $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ y un objeto distinguido $1 \in \mathbf{Ob}(\mathcal{C})$ que satisface las siguientes ecuaciones:

1. $1 \otimes f = f = f \otimes 1$
2. $f \otimes (g \otimes h) = (f \otimes g) \otimes h$

para cualesquiera $f, g, h \in \mathbf{Mor}(\mathcal{C})$
Un funtor monoidal estricto es un funtor entre categorías monoidales $F : \mathcal{C} \to \mathcal{D}$ que preserva el producto tensorial, es decir, $F(g \otimes_{\mathcal{C}} f) = F(g) \otimes_{\mathcal{D}} F(g)$

Notemos que en una categoría monoidal nuestros morfismos pueden componerse de dos maneras. Dado $f : A \to B$ y $g : B \to C$ morfismos podemos construir la composición $g \circ f : A \to C$. Pero también dados $u : A \to C$ y $v : B \to D$ podemos usar nuestro funtor para formar el morfismo $u \otimes v : A \otimes B \to C \otimes D$.

Es usual interpretar ambas composiciones de la siguiente manera: la composición usual es secuencial, es decir, primero se ejecuta el primer morfismo y después aplicamos el segundo; y la composición funtorial es paralela, es decir, ejecutamos ambos morfismos de manera simultánea.

[1] Un bifuntor es un funtor en cada entrada

Las consideraciones anteriores motivan un lenguaje gráfico para categorías monoidales mediante cables y cajas, recursivamente se define de la siguiente manera: Sean A, B, C, D objetos y $f : A \to B$, $g : B \to C$ y $h : C \to D$ morfismos en una categoría monoidal, tenemos la siguiente representación gráfica, llamada diagrama de cables:

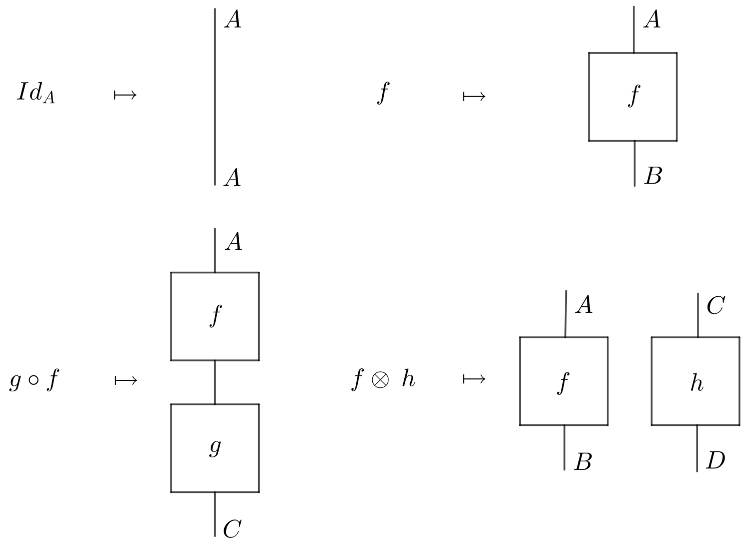


Figura 1: Lenguaje gráfico para categorías monoidales

El lenguaje anterior es tan poderoso que Joyal y Street demostraron que dos morfismos en una categoría monoidal son iguales (según los axiomas de categoría monoidal) si y sólo si los diagramas correspondientes son equivalentes. Dos diagramas son equivalentes si uno puede ser deformado, sin cruzar o desconectar cables, en el otro. Lo anterior es conocido como el teorema de coherencia para el cálculo gráfico en categorías monoidales.

Finalmente, es importante señalar que las dos composiciones son compatibles según las leyes de intercambio que enunciaremos a continuación.

Proposición 1. *Sea* $(\mathcal{C}, \otimes, 1)$ *una categoría monoidal y sean* $f : A \to B$, $g : B \to C$, $h : D \to E$ *y* $j : E \to F$ *morfismos de* $\mathcal{C}$*, entonces*

$$(g \circ f) \otimes (j \circ h) = (g \otimes j) \circ (f \otimes h) \tag{1}$$

$$(Id_B \otimes h) \circ (f \otimes Id_D) = f \otimes h = (f \otimes Id_E) \circ (Id_A \otimes h) \tag{2}$$

Demostración. Sean $f : A \to B$, $g : B \to C$, $h : D \to E$ y $j : E \to F$ morfismos en una categoría monoidal $\mathcal{C}$.
Primero probemos 1.

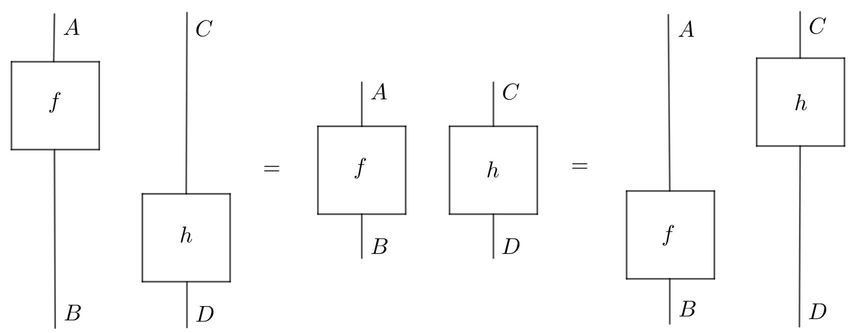
$$\begin{aligned}
(g \circ f) \otimes (j \circ h) &= \otimes(g \circ f, j \circ h) \\
&= \otimes((g, j)) \circ (f, h)) \\
&= (\otimes(g, j)) \circ (\otimes(f, h)) \qquad \text{por la bifuntorialidad de } \otimes \\
&= (g \otimes j) \circ (f \otimes h)
\end{aligned}$$

Ahora, veamos 2

$$
\begin{aligned}
(Id_B \otimes h) \circ (f \otimes Id_D) &= (Id_B \circ f) \otimes (h \circ Id_D) && \text{Por } 1 \\
&= f \otimes h \\
&= (f \circ Id_A) \otimes (Id_E \circ h) \\
&= (f \otimes Id_E) \circ (Id_A \otimes h) && \text{Por } 1
\end{aligned}
$$

Las justificaciónes también podrían escribirse al final, como ”donde la tercera igualdad se debe a la bifuntorialidad de $\otimes$” □

Podemos expresar en el lenguaje gráfico de categorías monoidales la ecuación 2 de la siguiente forma:

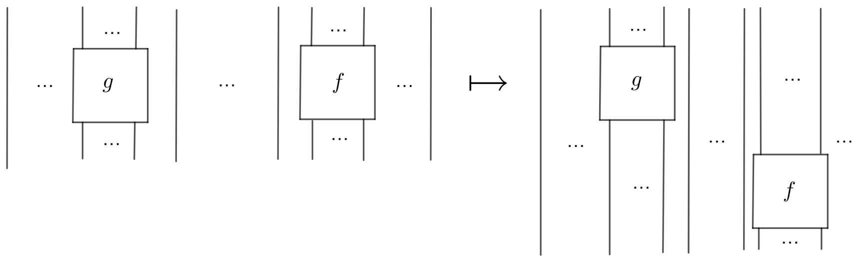


La ecuación anterior sugiere que siempre podemos encontrar diagramas donde cada caja este en un nivel distinto al resto, así que probémoslo, pero antes veamos la siguiente definición para poder enunciar nuestra proposición.

Definición 3. Un diagrama de cables está en posición general si a la derecha o izquierda de cada caja no hay otra.

Proposición 2. *Todo diagrama de cajas es equivalente a un diagrama de cajas en posición general.*

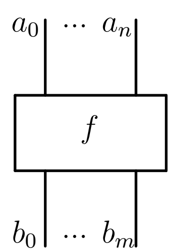
Demostración. Sea D un diagrama de cajas. Haremos la demostración por inducción sobre el número de cajas en el diagrama.
La hipótesis de induccion es trivial, pues si D tiene una caja ya acabamos.
Supongamos el resultado es cierto para algún n, y que D tiene $n+1$ cajas.
Sea f la caja ubicada más a la derecha y abajo en el diagramas. Consideremos D' el diagrama obtenido al sustituir f por $id_{dom(f)}$. Entonces D' tiene n cajas y por hipótesis es equivalente a un diagrama de cajas G' en posición general, consideremos ahora el diagrama G que se obtiene al pegarle f en el cable al que se lo habíamos cortado. Observemos que (1) la caja que colocamos sigue siendo la más derecha y abajo, y (2) a su izquierda hay a lo más una caja g (si no, ya acabamos), pues G' está en posición general. En ese caso, basta hacer la siguiente deformación:



□

La proposición anterior será una parte fundamental en nuestro próximo objetivo: generar una categoría monoidal libre a partir de una signatura monoidal. Para ello, primero describiremos un lenguaje gráfico para las signaturas monoidales pues lo usaremos en la descripción de categorías monoidales.

Sea G una signatura monoidal y sea $f : \vec{a} \to \vec{b} \in G_1$. Diremos que f es una caja y la representaremos como:



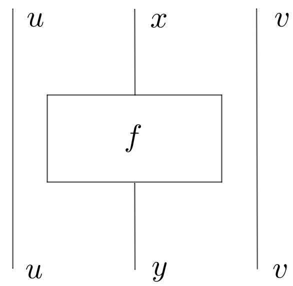
Además, tenemos dos casos distinguidos, cuando $\vec{a}$ o $\vec{b}$ son la palabra vacía, en esos casos diremos que f es un estado o un efecto respectivamente y lo denotaremos como

Usaremos la descripción anterior para construir la categororía monoidal libre $\mathbf{MC}(G)$ asociada a una signatura monoidal, para ello primero veamos qué es una signatura de niveles.

Dada G una signatura monoidal definimos su signatura de niveles como

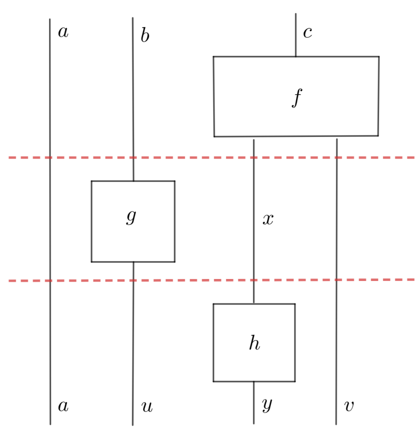
$$G_0^* \xleftarrow[\texttt{cod}]{} L(G) = G_0^* \times G_1 \times G_0^* \xrightarrow{\texttt{dom}} G_0^*$$

donde para cada nivel $l = (u, f : x \to y, v)$ en $L(G)$ definimos $\mathtt{dom}(l) = uxv$ y $\mathtt{cod}(l) = uyv$, gráficamente representamos a l de la siguiente manera



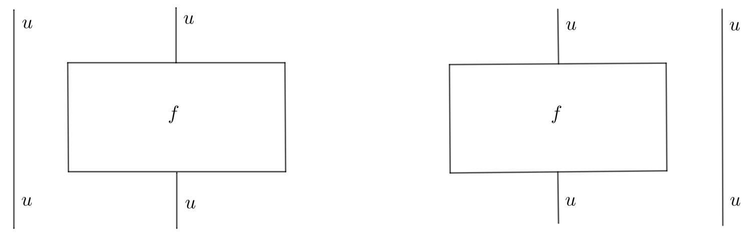
Consideremos ahora $\mathbf{PMC}(G)$, el conjunto de diagramas premonoidales, como el conjunto de morfismos de la categoría libre generada por la signatura de niveles de G. Notemos que estos morfismos son secuencias de niveles de la forma $(d_1, \ldots, d_n)$ tales que $\mathtt{cod}(d_i) = \mathtt{dom}(d_{i+1})$ para cada $i \in \{1, ..., n-1\}$.

Veamos el siguiente ejemplo de un diagrama premonoidal generado con la signatura $G_0 = \{a, b, c, u, v, y, x\}$ y $G_1 = \{f : c \to xv, g : b \to u, h : x \to y\}$



Supongamos que queremos construir el diagrama a partir de la signatura, ¿qué información necesitamos? En principio, conocer el número de niveles que conforma el diagrama, en nuestro caso 3, luego conocer el dominio y codominio del diagrama, abc y $auyv$ respectivamente, además de la lista de cajas en cada nivel, $[f, g, h]$. Finalmente necesitamos conocer la posición en la que se encuentra la caja en cada nivel, lo cual lo identificamos señalando el número de cables a la izquierda de la caja, $(2, 1, 2)$

Aunque la última condición es innecesaria en el diagrama anterior, es importante en casos como el siguiente, donde los primeros datos no determinan un único diagrama premonoidal.



La explicación anterior motiva nuestra siguiente proposición, una manera eficiente de representar la información contenida en un diagrama premonoidal:

Proposición 3 (Codificación de diagramas premonoidales)**.** *Sea G una signatura monoidal. Un diagrama premonoidal d en $\boldsymbol{PMC}(G)$ está unicamente determinada de la siguiente forma:*

1. *un dominio:* $\texttt{dom}(d) \in G_0^*$
2. *un codominio:* $\texttt{cod}(d) \in G_0^*$
3. *una lista de cajas:* $\texttt{cajas}(d) \in G_1^n$
4. *una lista de números identificadores:* $\texttt{id}(d) \in \mathbb{N}^n$

donde n es el número de niveles del diagrama d y el $n-$ésimo identificador señala la posición de la caja en el $n-$ésimo nivel indicando el número de "cables" a la izquierda de la caja.
Más aún, la información anterior define un diagrama premonoidal válido si para cada $\imath \in \{1, ..., n\}$ tenemos que

$$\texttt{peso}(d)_i \geq \texttt{id}(d)_i + |\texttt{dom}(b_i)|$$

donde

$$\texttt{peso}(d)_1 = |\texttt{dom}(d)| \qquad \texttt{peso}(d)_{i+1} = \texttt{peso}(d)_i + |\texttt{cod}(b_i)| - |\texttt{dom}(b_i)|$$

con $b_i = \texttt{boxes}(d)_i$

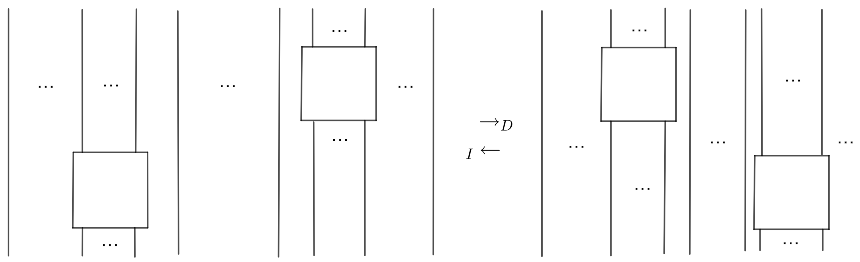
Notemos que la función `peso` indica el número de cables con los que inicia cada nivel, así la desigualdad significa que este número debe ser mayor al número de cables que estarán a la izquierda de la caja menos el número de cables que necesita la caja del nivel. Bajo estas condiciones, es evidente que el diagrama generado será válido.

Ahora bien, notemos que los elementos de $\mathbf{PMC}(G)$ se parecen mucho a diagramas de cables en posición general, sin embargo, necesitamos establecer cuando dos diagramas premonoidales son equivalentes, para ello veamos las siguiente definiciones.

Definición 4. Sea G una signatura y d un diagrama premonoidal de G. Decimos que d admite un intercambio izquierdo en el nivel n si $\texttt{id}(d)_{n+1} \geq \texttt{id}(d)_n + |cod(b_n)|$ y admite un intercambio derecho en el nivel $n+1$ si $\texttt{id}(d)_n \geq \texttt{id}(d)_{n+1} + |dom(b_n)|$.

En otras palabras, un intercambio es posible cuando las cajas a intercambiar no tienen cables en común. Ahora sí, veamos que es un intercambio

Definición 5. Sea G una signatura y d un diagrama premonoidal de G. Si d admite un intercambio izquierdo (o derecho) en el nivel n, entonces generamos un nuevo diagrama idéntico en todos los niveles a d salvo en n y $n+1$ donde hay un cambio de la siguiente forma:



decimos que el nuevo diagrama el intercambio izquierdo (o derecho) en el nivel n de d.

Claramente los diagramas producidos tras un intercambio es también un diagrama válido. Definimos una reducción de diagramas como una secuencia de intercambios izquierdos o derechos. Con ello, definimos la siquiente relación: dos diagramas d y d' están $\sim$-relacionados si existe una reducción que lleve d a d'.

Proposición 4. *Sea G una signatura monoidal. La relación $\sim$ sobre $\boldsymbol{PMC}(G)$ es de equivalencia.*

Demostración. Sea G una signatura monoidal. Sea d una diagrama premonoidal. Consideramos la secuencia de reducciones vacía, entonces $d \sim d$. Por lo tanto, $\sim$ es reflexiva.
Sea d y d' diagramas tales que $d \sim d'$. Sea $p_1^{r_1}, p_2^{r_2}, ..., p_n^{r_n}$ la secuencia de reducciónes que lleva d a d' con $r_i \in \{d, i\}$ para todo $i\{1, ..., n\}$, es decir, indica si el intercambio fue izquierdo o derecho. Notemos que un intercambio izquierdo es el inverso del derecho y viceversa, entonces consideremos r_i' como el opuesto a r_i. Entonces la reducción $p_n^{r_n'}, p_{n-1}^{r_{n-1}'}, ..., p_1^{r_1'}$ lleva d' a d. Por lo tanto, $d' \sim d$. Por lo tanto, $\sim$ es simétrica.
Por último, sean d, d' y d'' tales que $d \sim d'$ y $d' \sim d''$. Entonces hay una reducción que llevar d a d' y una que lleva d' a d'', así hay una reducción que lleva d a d'', es decir, $d \sim d''$. Por lo tanto, $\sim$ es transitiva.
Por lo tanto, $\sim$ es de equivalencia. $\square$

Otra forma de expresar la proposición anterior es señalando que cualesquiera dos diagramas premonoidales tales que pueda deformarse, sin cruzar o cortar cables, uno en el otro están $\sim$-relacionados. Con todo lo anterior, ya estamos en condiciones de conocer a nuestra categoría monoidal libre, $\mathbf{MC}(G)$.

Proposición 5. *Sea G una signatura monoidal, entonces:*

$$\boldsymbol{MC}(G) \cong \boldsymbol{PMC}(G)/\sim$$

Demostración. Sea G una signatura monoidal.
Notemos que cualquier combinación con sentido de elementos de G_1 con $\circ$ y $\otimes$ puede expresarse como un diagrama de cables, el cual a su vez puede transformarse a un diagrama equivalente en posición general en virtud de la proposición 2. A su vez, todo diagrama en posición general puede ser expresado como un diagrama premonoidal, por lo tanto, cualquier morfismo en $\mathbf{MC}(G)$ se encuentra en $\mathbf{PMC}(G)$.
Por otro lado, la proposición 4 nos dice que cualquiera dos diagramas premonoidales tales que pueda deformarse, sin cruzar o cortar cables, son iguales. Pero de acuerdo al teorema de coherencia para el cálculo gráfico podemos concluir que $\mathbf{PMC}(G)/\sim$ satisface los axiomas de una categoría monoidal. $\square$

0.2. Gramáticas monoidales

Veamos como podemos darle una interpretación lingüística a una signatura monoidal G. Podemos considerar los objetos de $\mathbf{MC}(G)$ como cadenas de símbolos de G_0, las cajas (o flechas generadores) de G_1 como reglas de producción y los morfismos en $\mathbf{MC}(G)$ como derivaciones. Formalmente podemos definir un 'lenguaje monoidal' de la siguiente manera:

Definición 6. Una gramática monoidal es una signatura monoidal finita G de la siguiente forma

$$(V+B)^* \xleftarrow{\texttt{dom}} G \xrightarrow{\texttt{cod}} (V+B)^*$$

donde V es un conjunto de palabras llamada vocabulario y B es un conjunto de símbolos con uno inicial $s \in B$, llamado el símbolo de oración.
Una oración $u \in V^*$ es gramaticalmente correcta si hay un morfismos $g : u \to s$ en $\mathbf{MC}$(G). Definimos el lenguaje generado por G como:

$$\mathcal{L}(G) = \{u \in V^* | \exists g : u \to s \text{ en } \mathbf{MC}(G)\}$$

A los lenguajes generados por gramáticas monoidales les llamamos lenguajes monoidales.

Ejemplo 1. Sabemos que los lenguajes libres de contextos son producidos por gramáticas de hipergráficas de la siguiente forma:

$$(V+B)^* \leftarrow G \to B$$

Ya que $B \subset (V+B)^*$, entonces son signaturas monoidales y, por lo tanto, es un lenguaje monoidal.

Tal como se prometió al inicio del capítulo, veremos que los lenguajes monoidales son exactamente los lenguajes recursivamente enumerables. Pero antes convendrá recordar las dos definiciones equivalentes, la primera en términos de las gramáticas que los generen y la segunda en términos del modelo de cómputo que los aceptan.

Definición 7. Un gramática de tipo 0 (o sin restricciones) es una tupla $G = (V, B, P, s)$ donde V es un vocabulario, B es un conjunto de símbolos no terminales con un símbolo distinguido s y P es un conjunto de reglas de producción de la forma $\alpha \to \beta$ donde $\alpha, \beta \in (V + B)^*$.
Si $\alpha \to \beta$ es una regla de producción, para cualquier $\gamma, \delta \in (V + B)^*$ definimos la derivación

$$\gamma\alpha\delta \Rightarrow \gamma\beta\delta$$

Definimos el lenguaje generado por G como:

$$\mathcal{L}(G) = \{v \in V^* | s \Rightarrow^* v\}$$

donde $\Rightarrow^*$ es la cerradura reflexiva y transitiva de $\Rightarrow$.

Teorema 1. *Los lenguajes monoidales son exactamente los generados por una gramática de tipo 0.*

Demostración. Primero veamos que todo lenguaje monoidal es producido por una gramática de tipo 0.
Sea G una gramática monoidal.
Para cada $f : u \to v$ en G_1 definimos la regla de producción $p_f : v \to u$ en P con $v, u \in (V + B)^*$. Notemos que si existen dos flechas generadoras $f, g : u \to v$, entonces $p_f = p_g$.
Sea $w \in \mathcal{L}(G)$, entonces existe $f : w \to s$ en $\mathbf{MC}(G)$, eso implica existen flechas generadoras tal que el siguiente diagrama conmuta

$$\begin{tikzcd} w \arrow[r, "f_1"] \arrow[rr, bend right] & \dots \arrow[r, "f_n"] & s \end{tikzcd}$$

$$f$$

Entonces tenemos la siguiente derivación de P:

$$s \Rightarrow \cdots \Rightarrow w$$

Es decir, tenemos que $s \Rightarrow^* w$ y, por lo cual, w está en el lenguaje generado por la grámatica de tipo 0 (V, B, P, s). Por lo tanto, $\mathcal{L}(G) \subset \mathcal{L}(V, B, P, s)$. La otra contención se da leyendo el argumento anterior al revés y, por lo tanto, $\mathcal{L}(G) = \mathcal{L}(V, B, P, s)$.
Resta probar que todo lenguaje generado por una gramática de tipo 0 es monoidal.
La idea es análoga anterior y, dado que hemos realizado pruebas semejantes en los capítulos anteriores, entonces nos tomaremos la libertad de omitir esta demostración. □

Ejemplo 2.

Como se acaba de comentar, la prueba fue sencilla en virtud de la semejanza entre las definiciones de gramática monoidal y gramática de tipo 0. Será más divertido ver como se relaciona nuestra definición categórica con el modelo de computo asociado. Para esto, veamos alguna definiciones preliminares:

Definición 8. Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$ una máquina de Turing.
Definimos un movimiento de la siguiente manera: Dado $X_1X_2 \cdots X_{i-1}qX_i \cdots X_n$ en la banda y $\delta(q, X_i) = (p, Y, \leftarrow)$ tenemos que:

$$X_1X_2 \cdots X_{i-1}qX_i \cdots X_n \vdash_M X_1X_2 \cdots X_{i-2}pX_{i-1}Y \cdots X_n$$

Pero si $\delta(q, X_i) = (p, Y, \rightarrow)$, entonces

$$X_1X_2 \cdots X_{i-1}qX_i \cdots X_n \vdash_M X_1X_2 \cdots X_{i-1}YpX_{i+1} \cdots X_n$$

Definimos $\vdash_M^*$ como la cerradura reflexiva y transitiva de $\vdash_M$.
El lenguaje M aceptado por una máquina de Turing se define como

$$\mathcal{L}(M) = \{w \in \Sigma^* | q_0w \vdash_M^* \alpha s \beta \text{ con } s \in F \text{ y } \alpha\ ,\ \beta \in \Gamma^*\}$$

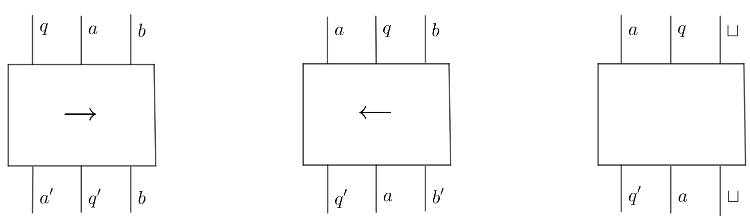
Es decir, el lenguaje producido por una máquina de Turing es el conjunto de cadenas del vocabulario tales que al inicializarse la banda con esas cadenas logramos alcanzar algún estado final. Ya estamos en condiciones de dar la siguiente definición:

Definición 9. Un lenguaje L es recursivamente enumerable si existe una máquina de Turing M que acepta L, es decir, $L = L(M)$.

Un resultado bien conocido es que que los lenguajes recursivamente enumerables tienen el mismo poder expresivo que los lenguajes generados por una gramática de tipo 0. Veamos, entonces, la manera d

Proposición 6. *Para todo lenguaje recursivamente enumerable L, existe una gramática monoidal G tal que $L = \mathcal{L}(G)$*

Demostración. Sea L un lenguaje recursivamente enumerable generado por la máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$.
Definimos $B = (\Gamma \setminus \Sigma) + Q$ y $V = \Sigma$ con $s \in B$ el símbolo de oración. Las reglas de producción son las siguientes:



donde $a, a', b, b' \in \Sigma$, $q, q' \in Q$ tales que $\delta(q, a) = (q', a', \rightarrow)$, $\delta(q, b) = (q', b', \leftarrow)$ y $\delta(q, \sqcup) = (q', \sqcup, \rightarrow)$. Además, agregamos las reglas de producción $x \rightarrow q_0x$ y $xsy \rightarrow s$, donde la segunda sirve para "limpiar la banda".
Veamos que $L = \mathcal{L}(G)$
Sea $w \in L$. Entonces $q_0w \vdash_M^* \alpha s \beta$. Pero notemos que para cada aplicación $\vdash_M$ tenemos un morfismo que actúa de la misma manera, entonces tenemos morfismos tales que

$$q_0w \xrightarrow{f_1} \ldots \xrightarrow{f_n} \alpha s \beta$$

Además como también tenemos las reglas de producción $w \to q_0w$ y $\alpha s\beta \to s$ tenemos una derivación $w \to s$ en G. Por lo tanto, $w \in \mathcal{L}(G)$
Para el regreso, sea $w \in \mathcal{L}(G)$, entonces existe una derivación $w \to s$.
Observemos que por las reglas producción, tenemos la derivación es necesariamente de la siguiente forma

$$w \longrightarrow q_0w \longrightarrow ... \longrightarrow \alpha s\beta \longrightarrow s$$

Pero la derivación intermedia solo puede lograrse con reglas de producción de la forma $\leftarrow$ y $\rightarrow$, las cuales corresponden con derivaciones en $\vdash_M$, por lo tanto, $q_0w \vdash_M^* \alpha w\beta$.
Por lo tanto, $w \in L$.
Por lo tanto, $L = \mathcal{L}(G)$ □

Terminemos este capítulo con el siguiente ejemplo

Ejemplo 3. El lenguaje $L = \{a^ib^ic^i|i \in \mathbb{N}\}$ no es regular, pero sí es recursivamente enumerable.
Una máquina de Turing que acepte el lenguaje anterior es la siguiente:

δ	a	b	c	x	y	z	$\sqcup$
q_0	$q_1, x, \rightarrow$				$q_4, y, \rightarrow$		$q_5, \sqcup, \rightarrow$
q_1	$q_1, a, \rightarrow$	$q_2, y, \rightarrow$			$q_1, y, \rightarrow$		
q_2		$q_2, b, \rightarrow$	$q_3, z, \leftarrow$			$q_2, z, \rightarrow$	
q_3	$q_3, a, \leftarrow$			$q_0, x, \rightarrow$		$q_3, z, \leftarrow$	
q_4					$q_4, y, \rightarrow$	$q_4, z, \rightarrow$	$q_5, \sqcup, \rightarrow$
q_5							$q_5, s, \rightarrow$

Veamos la derivación en gramática monoidal correspondiente de la palabra abc